

Morphology-based Noise Reduction: Structural Variation and Thresholding in the Bitonic Filter

Graham Treece

Abstract—The bitonic filter was recently developed to embody the novel concept of signal bitonicity (one local extremum within a set range) to differentiate from noise, by use of data ranking and linear operators. For processing images, the spatial extent was locally constrained to a fixed circular mask. Since structure in natural images varies, a novel structurally varying bitonic filter is presented, which locally adapts the mask, without following patterns in the noise. This new filter includes novel robust structurally varying morphological operations, with efficient implementations, and a novel formulation of non-iterative directional Gaussian filtering. Data thresholds are also integrated with the morphological operations, increasing noise reduction for low noise, and enabling a multi-resolution framework for high noise levels. The structurally varying bitonic filter is presented without presuming prior knowledge of morphological filtering, and compared to high-performance linear noise-reduction filters, to set this novel concept in context. These are tested over a wide range of noise levels, on a fairly broad set of images. The new filter is a considerable improvement on the fixed-mask bitonic, outperforms anisotropic diffusion and image-guided filtering in all but extremely low noise, non-local means at all noise levels, but not the block-matching 3D filter, though results are promising for very high noise. The structurally varying bitonic tends to have less characteristic residual noise in regions of smooth signal, and very good preservation of signal edges, though with some loss of small scale detail when compared to the block-matching 3D filter. The efficient implementation means that processing time, though slower than the fixed-mask bitonic filter, remains competitive.

Index Terms—bitonic filter, morphology, noise reduction, edge preservation

I. INTRODUCTION

THERE are many situations in which a corrupting additive component (noise) needs to be removed from an *a priori* unknown digital signal, for instance a two-dimensional image. Whilst this problem has no universal solution, natural and synthetic images have features which are sufficiently different from typical noise to enable the development of many practically useful noise-reduction algorithms. The performance and characteristics of such algorithms are critically dependent on how they each define the difference between ‘noise’ and ‘signal’. Such definitions are wide-ranging, for instance based on transformed domains (noise has higher frequency), data level (signal has greater difference in data value), pattern-matching (signal structure is more repeatable) or even the historical record (signal is previously known).

In this context, the novel concept of using signal ‘bitonicity’ for differentiation was recently proposed [1], where the signal is deemed to be anything containing one maximum

or one minimum over a given spatial range. This definition is crucially independent of data value, being based instead on data *ordering* or *rank*: hence it applies equally to both smooth and disjoint signals. The bitonic filter was developed as a combination of rank-based (robust morphological openings and closings) and linear (Gaussian filtering) operators, in order to remove non-bitonicity (noise) from corrupted signals, with an initial application in medical computed tomography [2]. It was shown to have good noise-reduction performance across a range of noise levels, surpassing other morphology-based alternatives, and even competing in some cases with more well known linear filters such as anisotropic diffusion [3] or non-local means [4], particularly for images with varying noise levels.

Whilst, for one-dimensional signals, the spatial range over which bitonicity is imposed is defined purely by the filter window length, for two-dimensional (2D) images, a 2D structuring element or ‘mask’ defines which image pixels locally contribute to the data ranking. In the bitonic filter, this mask was fixed to a circle with a chosen diameter. Since the mask shape imposes a structure on the signal, and yet the structure of the signal is not expected to be constant over an image, it is clear that allowing the shape to vary locally could offer significant performance improvements.

This concept of structurally varying morphological operations (also known as adaptive morphology) has been the subject of recent research [5] with well developed mathematical foundations [6]. There are a variety of ways to change the shape of the mask, but the extent to which the mask is allowed to shape itself to the data is related to the noise reduction performance [5]. Strong prior definitions of shape are required if high noise reduction is desired, since otherwise the mask can conform to the noise as well as the signal. Pre-defined flat masks [7] are hence the most appropriate for inclusion in the bitonic filter, with ellipses the simplest extension from a circle. Having defined the type of structurally varying mask, its orientation and specific shape needs to be able to adapt to the image data. The orientation is usually derived from the gradient, often in the form of a structure tensor [7], [8], [9] also called the average squared gradient [10]. The local anisotropy can be used to control the shape, for instance setting the aspect ratio of an ellipse [7].

It is the aim of this paper to develop the bitonic filter so that it can make use of structurally varying masks. Structural variation is a relatively new development within mathematical morphology, and, other than the median filter, mathematical morphology itself is a less widely used technique than linear image filtering, perhaps due to complexity in implementation and analysis. Hence the paper covers sufficient detail to show

how the structurally varying bitonic can be implemented efficiently, without presuming familiarity with the field. In order to develop this filter, several extensions to structurally varying morphology are also proposed. Firstly, the bitonic filter unusually involves robust (not involving maxima or minima) morphological openings and closings, and efficient implementations of such operations do not exist. In addition, a better definition of mask orientation is developed, making use of trials over multiple masks rather than solely relying on the structure tensor. The incorporation of data thresholds, and the inclusion of the whole in a multi-resolution framework, is also demonstrated. For the new bitonic filter, an alternative to the Gaussian filter which can structurally match the varying morphological operations is also required.

Since the bitonic filter is a recent concept, its performance is placed in context by comparison with a wide range of more well known linear filters, on a range of images, across multiple levels of noise. It is seen that the structurally varying bitonic can achieve competitive noise reduction, with practical processing times, across the entire noise range, though with particularly promising results in very high noise.

II. METHODS

The bitonic filter was analysed in detail in [1]. Briefly, it consists of a robust opening $O_{w,c}$ and closing $C_{w,c}$ of a signal $I(\mathbf{x})$ at location \mathbf{x} :

$$R_{w,c}(I(\mathbf{x})) = c^{\text{th}}_{\mathbf{y} \in w} \text{centile} \{I(\mathbf{x} + \mathbf{y})\} \quad (1)$$

$$O_{w,c} = R_{w,100-c}(R_{w,c}(I(\mathbf{x}))) \quad (2)$$

$$C_{w,c} = R_{w,c}(R_{w,100-c}(I(\mathbf{x}))) \quad (3)$$

where $R_{w,c}$ is a rank filter, and \mathbf{y} is a vector distance to a location close to \mathbf{x} , within a filter region w (or mask in 2D). This ranks (sorts) I over w and returns the intensity corresponding to the chosen centile c from the sorted list. The window size (or number of elements in 2D) is given by $|w|$. The centile c is usually set to 10% for a bitonic filter with a fixed circular mask. Setting $c = 100\%$ in eq. (1) would return the maximum value (known as a dilation) and $c = 0\%$ the minimum (known as an erosion), but in practice a small non-zero centile value as in [11] gives more robust results in the presence of noise.

Opening and closing operations are not self-dual (symmetric in data value), and do not preserve the mean signal intensity. To correct this defect, the operations are weighted, by considering their difference from the original signal. This difference is filtered with a Gaussian linear filter, $G_\sigma(\mathbf{x})$, with standard deviation $\sigma = 0.33l$ where l is the diameter of the mask in 2D:

$$\epsilon_O = |G_\sigma(I(\mathbf{x}) - O_{w,c})| \quad (4)$$

$$\epsilon_C = |G_\sigma(C_{w,c} - I(\mathbf{x}))| \quad (5)$$

$$b_{w,c} = \frac{\epsilon_O C_{w,c} + \epsilon_C O_{w,c}}{\epsilon_O + \epsilon_C} \quad (6)$$

where $b_{w,c}$ is the output of the bitonic filter. The opening and closing operations effectively detect bitonic signals, and hence ϵ_O and ϵ_C represent smoothed versions of the residual errors

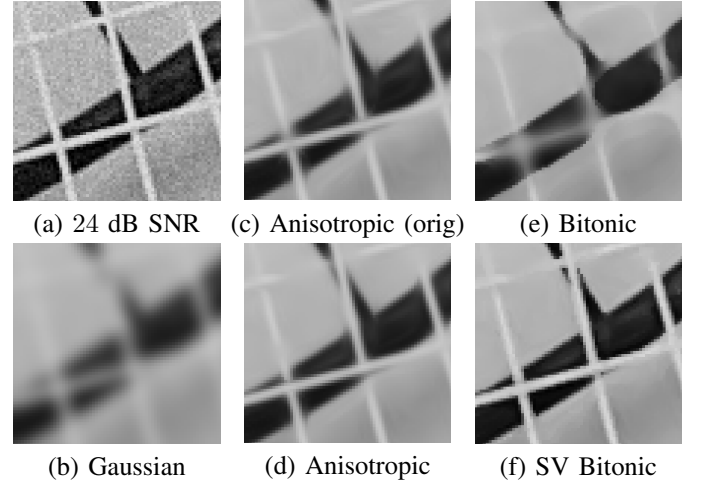


Fig. 1. Development of the structurally varying (SV) bitonic filter. (a) Section of a noisy image, filtered in all other images with the same mask size. (b) Gaussian filter, also used in the fixed bitonic (e). (c) Anisotropic filter, with (d) improved corner response, used in the SV bitonic. (f) SV bitonic, with mask selection based on anisotropic filtering and morphological operations.

after removing such signals from the original. The result is to preserve bitonic signals, but reduce noise in all regions, including across signal edges.

A slightly different formulation of eq. (6) is possible:

$$b_{w,c} = \frac{\epsilon_O^m (C_{w,c} - \epsilon_C) + \epsilon_C^m (O_{w,c} + \epsilon_O)}{\epsilon_O^m + \epsilon_C^m} \quad (7)$$

where m controls the transition between $C_{w,c}$ and $O_{w,c}$. Setting m to 1 gives a gradual transition, and results in the same expression as in eq. (6): $m = 3$ gives a more sudden transition, which slightly improves the performance, and is used in the remainder of this paper. A typical result of this bitonic filter is shown in Fig. 1(e) for the noisy image in Fig. 1(a). For comparison, the result of applying $G_\sigma(I(\mathbf{x}))$ alone is in Fig. 1(b).

The following sections cover the extension of eq. (7) to the structurally varying (SV) version. Section II-A discusses the choice of masks, with appropriate centiles, and the efficient implementation of robust SV versions of the openings and closings in eqns. (2) and (3). Section II-B considers the incorporation of data thresholds into these operations, and the subsequent extension of eq. (7) to a multi-resolution framework. Section II-C develops an appropriate replacement for the Gaussian filters in eqns. (4) and (5).

Section II-D explores better techniques for selecting the mask shape and direction, and also addresses the practical choices when applying such a technique to colour images. Finally, Section II-E shows how these can be combined with the filtering and morphological operations in the previous two sections to implement an efficient SV bitonic filter. The outputs of these stages are summarised in Fig. 1.

A. Structurally varying robust morphological operations

1) *Design of mask sets:* Ellipses of varying orientation and aspect ratio have been used in SV closing and opening operations before [7], with a flexible definition of shape in terms

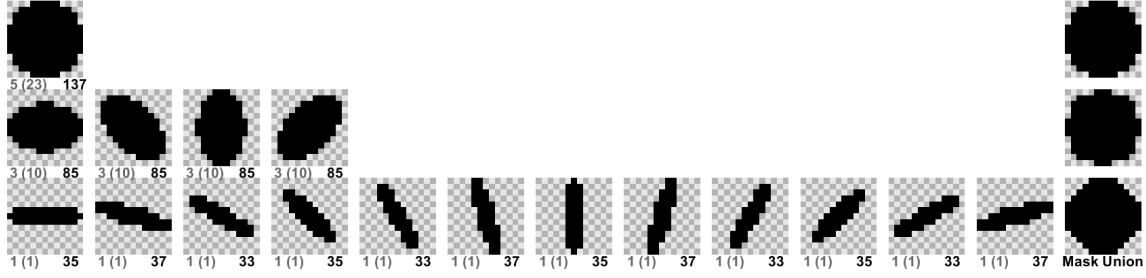


Fig. 2. Masks for SV morphology. This is the set of 17 masks for $r = 6, l = 13$, in which case there are 3 shapes ($s_{\max} = 2$), one per row, with $s = 0, a_0 = \frac{1.7}{8}$ corresponding to the bottom row, and $s = 2, a_2 = 1$ to the top row. The number of orientations, one per column, varies for each shape, with $o_{\text{tot},0} = 12$ for the thinnest ellipses, $s = 0$. The numbers below each mask are the rank (from 0) for the equivalent centile c_e (as ranks rather than percentages), followed by the adjusted c_s from eq. (34), followed by the total number of pixels $|w_s|$ in each mask. The three images on the right show the region covered by the union of the masks in each row.

of the two radial parameters. Implementation is improved if the number of masks is restricted, and in any case there is little point using masks which cover nearly identical sets of pixels. Hence the number of orientations for each shape varies with the aspect ratio of that shape. More are required for thin ellipses, since the percentage overlap is smaller for a given change in orientation, whereas for a circle, only one is needed. The maximum radius of the masks is a positive integer r , such that the number of pixels along one side of the square region containing all masks is $l = 2r + 1$. More variations in shape (aspect ratio) are possible for larger mask regions l . The maximum shape number s_{\max} , and aspect ratio a_s and number of orientations $o_{\text{tot},s}$ for each shape s , are given by:

$$s_{\max} = \min \left\{ \left\lceil \frac{r}{4} \right\rceil, 1 \right\} \quad (8)$$

$$a_s = \frac{1.7 + (r + 0.3) \frac{s}{s_{\max}}}{r + 2} \quad (9)$$

$$o_{\text{tot},s} = \max \{ 4 \lceil 0.4 (1 - a_s) (r + 2) \rceil, 1 \} \quad (10)$$

where $s = 0$ represents the thinnest ellipse, and $s = s_{\max}$ a circle. The various constants in these equations are designed to produce a sensible set of masks for any radius r , based purely on the geometry of each mask. In particular, the shapes s_{\max} and orientations $o_{\text{tot},s}$ balance the number of masks with the difference in overlap between them. The aspect ratio of each shape a_s ensures that the thinnest ellipse is always at least three pixels wide, since thinner lines tend to amplify noise characteristics in the image, as has been noted before [7].

Pixels in the $l \times l$ region are deemed to be in each mask if:

$$\left(\frac{r_{\text{major}}}{r + 0.25} \right)^2 + \left(\frac{r_{\text{minor}}}{a_s (r + 0.25)} \right)^2 \leq 1 \quad (11)$$

where r_{major} and r_{minor} are the distance of a particular pixel from the centre of the region, respectively along the major and minor axes of the appropriately oriented ellipse. These are both reduced by 0.25 (with a minimum of 0) in order to make the sides of each ‘ellipse’ straighter, particularly for thinner shapes. This apparently minor detail is important, since the exact shape of these masks, for instance the avoidance of single-pixel protrusions, has a noticeable effect on the output of the SV operations. An example for $r = 6$ is given in Fig. 2, in which case $s_{\max} = 2$, $a_s = \{ \frac{1.7}{8}, \frac{4.85}{8}, 1 \}$ and $o_{\text{tot},s} = \{ 12, 4, 1 \}$.

2) *Equivalent centiles for different mask shapes:* For the fixed circular mask, a centile $c = 10\%$ is always a good choice [1]. For SV masks with thinner shapes, any of the corresponding orientations could be chosen for processing a particular pixel in an image. As shown in Section II-D, one way to select the appropriate orientation is based on which mask fits the data best. In an image which only contains noise, there is no underlying structure and hence there should be no preference between mask shapes. However, if all masks used the same centiles, the thinner masks would seem to fit the data better simply because there are more orientations to try: effectively these thin masks would over-fit to apparent structure in the noise.

Hence larger centiles need to be used for fatter masks with fewer possible orientations. The exact relationship between mask shape, the centile for the thinnest mask c_0 , and the required additional centile, is complex, hence the experimental approach in Appendix A has been adopted. Whilst this addresses all values of c_0 , in practice $c_0 = 4\%$ always gives good results. In this case the compensated centiles for the specific shapes in Fig. 2 with $r = 6$ are $c_s = \{ 4, 12, 17 \}$. It should be emphasised that, though compensation of the centiles is important, the results are not particularly sensitive to the form of this compensation, and there are hence other alternatives to the experimentally-derived equations in Appendix A.

3) *Histogram-based and sorting-based implementations:* Ranking using fixed masks can be performed in nearly constant time with respect to the mask region size l [12], by keeping a sorted list as the mask moves over the image and only updating those values which change between locations. If the data is integer-valued, this list can be stored and updated efficiently as a histogram of data values, and the rank output at a given centile read from this histogram. For a fixed mask w , the two ranking operations in each opening or closing can be implemented identically, as in eq. (1).

For an SV opening or closing, the situation is more complex. In the first or forward pass (inner ranking in eq. (3)), the mask $w(\mathbf{x})$ is different at every location \mathbf{x} . Updating a sorted list is therefore no longer efficient, since there might be a dramatic difference in image pixels covered by each mask between two neighbouring locations. The reverse operation (outer ranking in eq. (3)) is fundamentally different: it is no longer the mask at the current pixel which matters, but whether the masks at

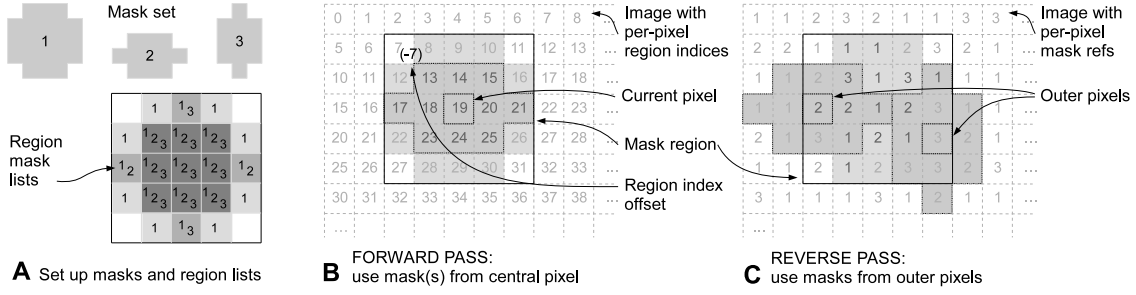


Fig. 3. Implementation of robust SV opening and closing. **A** Each mask is given a number (1 to 3 in the simple example above), and an array of width l , a square region bounding all the masks, is initialised with a list of mask intersections at each pixel. **B** Morphological operations proceed using the (fixed) union of all these masks (shown in light grey), which leads to very efficient ranking of all the image values within this area. An additional index is stored for each pixel, equal to $i \times l + j$, where (i, j) is the zero-based image pixel location. This is retained in the ranked list of image values, and preserves the location of the pixel even after ranking: subtracting the index from the top-left of the current region provides the location of each image value relative to the region mask lists. Hence the output of the forward ranking operation for different masks, with different centiles, can be calculated from the ranked superset. **C** For the reverse operation, ranking again starts using the union of the masks. The combination of the region index, and the per-image-pixel mask number from the forward operation, can be used to restrict the output to outer pixels from a mask that included the current pixel.

the surrounding pixels (used in the forward pass) contain the current pixel. Equations (1) to (3) are hence replaced with:

$$R_{w,c}(I(\mathbf{x})) = c^{\text{th}} \text{centile} \{I(\mathbf{x} + \mathbf{y})\}_{\mathbf{y} \in w(\mathbf{x})} \quad (12)$$

$$R_{w,c}^{-1}(I(\mathbf{x})) = c^{\text{th}} \text{centile} \{I(\mathbf{x} + \mathbf{y})\}_{-\mathbf{y} \in w(\mathbf{x} + \mathbf{y})} \quad (13)$$

$$O_{w,c} = R_{w,100-c}^{-1}(R_{w,c}(I(\mathbf{x}))) \quad (14)$$

$$C_{w,c} = R_{w,c}^{-1}(R_{w,100-c}(I(\mathbf{x}))) \quad (15)$$

where $R_{w,c}^{-1}$ denotes the ranking used in the reverse operation.

This reverse pass can be performed reasonably efficiently in cases where only the 0% (minimum) and 100% (maximum) centiles are being used, by refining the output in tandem with the forward pass [9]. Directional filtering, based on masks which are all lines, can also be implemented efficiently [8]. However, this latter technique does not adapt easily to other shapes, and neither technique is appropriate for robust operations using non-extreme centiles.

Hence a whole new approach is needed for a robust SV opening or closing with a fixed set of pre-determined masks. This approach, summarised in Fig. 3, is based on initially ranking the data using the union of all the masks, and then deducing the results for a particular mask at a given pixel from this superset of sorted data. The union mask can be ranked efficiently since it is fixed, and the additional region index preserves the original location of each sorted value. Any values in the superset of sorted data can then be correctly associated with each mask, and processed appropriately. Preserving the region index is easy for the sorting-based implementation, but has a greater detrimental effect on the histogram-based implementation. Whereas, for fixed masks, the histogram count at each data value was stored and either incremented (when adding a pixel to the sorted list) or decremented (when removing), for SV masks a separate list of region indices must be preserved at each data value, and the correct indices removed.

Figure 4 (a) and (b) contain forward and reverse examples with real data: the small squares in the overlay show the whole superset of sorted data, whilst the larger squares show the

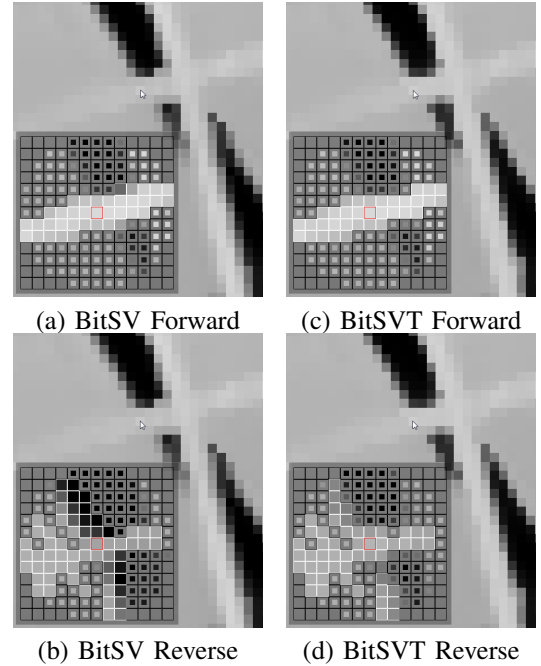


Fig. 4. Mask selection and ranking. The images show the result of a robust SV bitonic operation, with the bottom-left overlay containing the mask for the current pixel \mathbf{x} (denoted with an arrow). (a) In the forward operation, the union of possible masks (small squares) is ranked, but only data within the chosen mask $w(\mathbf{x})$ (larger squares) considered further. (b) In the reverse operation, data is considered whose masks $w(\mathbf{x} + \mathbf{y})$ cover the current pixel, and the effective mask is therefore more complex. (c) When using a threshold t , values beyond t from the current pixel are excluded from the mask in the forward operation. (d) In the reverse operation, values are only used if they derive from a mask centile which was not affected by such thresholding.

data which determines the rank output for the central pixel. It is apparent from this that, whilst the forward operation is recognisable as one of the thin ellipse masks, the reverse operation involves a far more complex mask shape which is related to coverage from the surrounding pixels.

B. Combining thresholds with morphological operations

1) *Implementation of relative threshold limits:* It is sometimes the case that the variance of the noise is reasonably well

known, or at least that an upper bound can be placed on the range of the noise. If such a bound can be estimated, then a threshold can be placed on the local range of data values, above which the data can no longer be attributed to noise alone. For the best performance in low noise, it is critical to make some use of this to preserve structural discontinuities above this level. Morphological operations do not make any specific use of such information, but the opening and closing operations can be adapted to include it.

In the forward SV operation, intensity values within the mask $w(\mathbf{x})$ are ignored if they are more than t different from the intensity at the current pixel, $I(\mathbf{x})$, hence eq. (12) becomes:

$$R_{w,c}(I(\mathbf{x})) = \underset{\mathbf{y} \in w(\mathbf{x}), |I(\mathbf{x}) - I(\mathbf{x}+\mathbf{y})| < t}{c^{\text{th}} \text{centile}} \{I(\mathbf{x} + \mathbf{y})\} \quad (16)$$

where t is the presumed noise range. This effectively changes the mask shape to ignore pixels whose data values are more different than can be explained by noise, as can be seen in the example of Fig. 4(c). In the reverse operation, eq. (13), pixels are only considered if their forward centile was *not* changed by this threshold, as in Fig. 4(d).

2) *Multi-resolution operation using thresholds:* Multi-resolution techniques, which entail processing different versions of an input image at different resolutions, are an obvious way to allow for larger masks without dramatically increasing the processing time. These can involve iterations on the residual image (the original minus the processed version) at a lower resolution, like a Laplacian pyramid [13]. However the residual contains very little of the image structure on which morphological operations rely, and hence processing of the residual does not make sense for such operations. A similar framework can still be used, but in this case the threshold t_n is lowered with each level n , and applied to the reduced, processed image, rather than the residual. The overall approach is shown in Fig. 5. Reduction of the image to quarter size (half in each dimension) is achieved with a restriction operator, as is typical in the multi-grid framework [14]: a Catmull-Rom spline [15] is a good choice, since it preserves edges as much as possible in the lower-resolution image. Expansion of the image is by a B-spline prolongation operator. Both these operators, denoted by diagonal arrows in Fig. 5, are very fast, but the combination is not lossless: to account for this, the prolonged, restricted image, without any further processing, is subtracted from the original before adding in the lower level results.

The threshold t_1 for the first level can be chosen according to the expected noise range, or set to the maximum data range (i.e. 256 for an 8-bit image). Subsequent thresholds t_n are calculated from the expected reduction in noise due to the SV bitonic filter applied at the previous level: this will be inversely proportional to filter radius r , since the noise reduces approximately with the size of the mask, hence:

$$t_n \approx \frac{t_{n-1}}{2.4r} \quad (17)$$

where the 2.4 in the denominator (which would be $\approx \sqrt{\pi}$ for a simple mean over a circular mask) accounts for the average SV mask shape and combination of operations in eq. (7). Three levels are typically sufficient, since t_n reduces quite rapidly.

These do not add much to the overall processing time, since although the filter radius r is constant (increasing the effective range of the filter at each level), images at lower levels have only one quarter of the pixels. This version of the multi-resolution SV bitonic is referred to as BitMV1.

Once the lower levels have been processed, the results are prolonged and added back into the previous result at the higher level. Some additional noise reduction can be achieved, particularly in high noise, by repeated application of the SV bitonic at this level, but with t_{n+1} , i.e. the threshold from the lower level. This option, referred to as BitMV2, will hence tend to double the processing time.

C. Linear filtering with structural variation

The fixed bitonic in eq. (7) employs a Gaussian filter to smooth the error and effectively give a mean image value with which to properly adjust the weights for the closing and opening operations. It is important that this filter extends over a similar domain to the morphological operations, so for the SV bitonic, a Gaussian-type filter is needed which will also vary in direction, but over a fixed size domain.

1) *Non-iterative anisotropic filtering:* The desired filtering direction can be calculated from the well known structure tensor T , or matrix of smoothed gradients $\{g_i, g_j\}$, in horizontal i and vertical j directions, of $I(\mathbf{x})$:

$$T = \begin{bmatrix} G_\sigma(g_i^2) & 2G_\sigma(g_i g_j) \\ 2G_\sigma(g_i g_j) & G_\sigma(g_j^2) \end{bmatrix} \quad (18)$$

$$\equiv \begin{bmatrix} T_{ii} & T_{ij} \\ T_{ij} & T_{jj} \end{bmatrix} \quad (19)$$

Following [16], the local direction $\phi(\mathbf{x})$ and degree of anisotropy $\gamma(\mathbf{x})$ can be derived from the eigenvalues $\lambda_{1,2}$ of T :

$$\lambda_{1,2} = \frac{1}{2} \left(T_{ii} + T_{jj} \pm \sqrt{(T_{ii} - T_{jj})^2 + 4T_{ij}^2} \right) \quad (20)$$

$$\gamma(\mathbf{x}) = 1 - \frac{\lambda_2}{\lambda_1} \quad (21)$$

$$\phi(\mathbf{x}) = \frac{1}{2} \tan^{-1} \left(\frac{T_{ij}}{T_{ii} - T_{jj}} \right)$$

where $\gamma = 0$ signifies low anisotropy (no dominant direction) and $\gamma = 1$ signifies high anisotropy (local gradients in only one direction). ϕ is then the angle following the dominant features in the image, i.e. the direction in which the filter should be aligned. The spatial range of γ and ϕ is determined by the extent of the Gaussian filter G_σ in eq. (18), which has $\sigma = 0.33l$ as before in order to match the mask region size $l \times l$. Figure 6(b) shows $\phi(\mathbf{x})$ for the image in Fig. 6(a) filtered with $l = 13$, and Fig. 6(c) the anisotropy $\gamma(\mathbf{x})$.

γ and ϕ can be used to construct a Gaussian-like filter $G_{\sigma,\alpha}(\mathbf{x})$ which follows the dominant direction, and whose

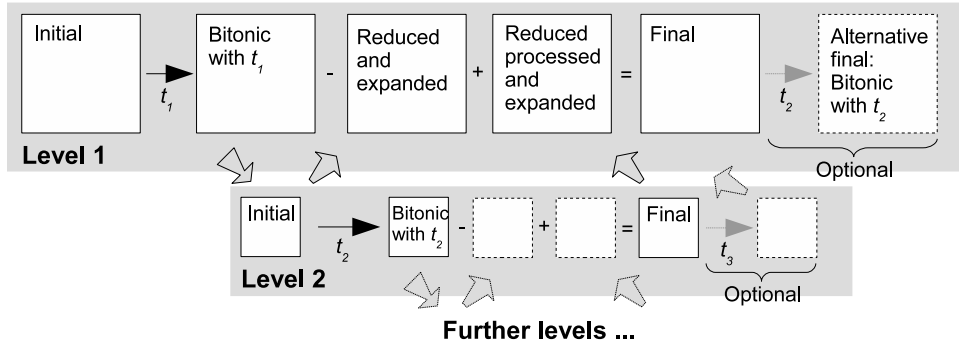


Fig. 5. Multi-resolution implementation of the bitonic filter using thresholds. At each level n , the bitonic filter operates on a quarter-size image, with a reduced threshold t_n , but with the same filter radius r in pixels, which means that the effective filter range increases. The threshold is reduced to account for the reduced noise level due to the bitonic filter previously applied at the higher level. Results from lower levels are expanded and added back in, taking into account the lossy nature of reduction (restriction) and expansion (prolongation). Optionally, the bitonic filter is then re-run at this level, but with an even lower threshold from the next level down.

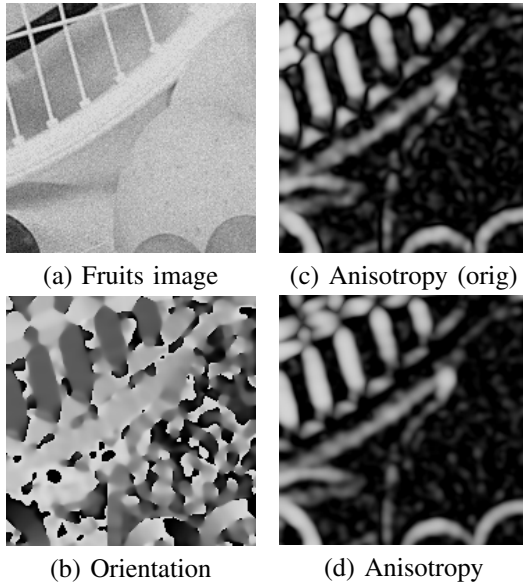


Fig. 6. Determination of local anisotropy. The extent and direction of anisotropy is shown for the image in (a). (b) The filtering direction is perpendicular to the largest eigenvector of the filtered structure tensor, with vertical as mid-grey and horizontal as white/black. (c) The anisotropy is derived from the difference between the maximum and minimum eigenvalues, with 1 represented as white and 0 as black. In corners or where lines cross, both eigenvalues are large and hence anisotropy is low, despite strong image gradients. (d) Improved anisotropy following eq. (25). See in particular the difference from (c) where the strings of the tennis racket cross (top left) and meet the frame (centre).

extent changes with anisotropy:

$$d_{\mathbf{y}}(\mathbf{x}) = |\mathbf{y} \sin(\angle(\mathbf{y}) - \phi(\mathbf{x}))| \quad (22)$$

$$\Omega_{\mathbf{y}}(\mathbf{x}) = \frac{e^{-\frac{|\mathbf{y}|^2}{2\sigma^2}}}{\left(\frac{d_{\mathbf{y}}(\mathbf{x})\gamma(\mathbf{x})^2}{\alpha^2} + 1\right) \left(\frac{d_{\mathbf{y}}(\mathbf{x}+\mathbf{y})\gamma(\mathbf{x}+\mathbf{y})^2}{\alpha^2} + 1\right)} \quad (23)$$

$$G_{\sigma,\alpha} = \frac{\sum_{\mathbf{y} \in w_l} \Omega_{\mathbf{y}}(\mathbf{x}) I(\mathbf{x} + \mathbf{y})}{\sum_{\mathbf{y} \in w_l} \Omega_{\mathbf{y}}(\mathbf{x})} \quad (24)$$

where \mathbf{x} is the current pixel location, and \mathbf{y} the vector distance to a neighbouring location within the rectangular window w_l of size $l \times l$. The numerator for the weighting $\Omega_{\mathbf{y}}(\mathbf{x})$,

eq. (23), is just a Gaussian in the distance between pixels $|\mathbf{y}|$, with standard deviation σ . However, it is reduced by the denominator which takes into account both the anisotropy $\gamma(\mathbf{x})$ and the dominant image direction, via $d_{\mathbf{y}}(\mathbf{x})$. This is the perpendicular distance between the pixel at $\mathbf{x} + \mathbf{y}$ and a line from \mathbf{x} drawn along the filtering direction $\phi(\mathbf{x})$, and hence has a small value if \mathbf{y} is in the same direction as given by $\phi(\mathbf{x})$. It is similar, at least in concept, to the geometric weight used in a trilateral filter [17]. Higher weights are associated with pixels along such a line (low $d_{\mathbf{y}}(\mathbf{x})$) or when the anisotropy is small (low $\gamma(\mathbf{x})$). The existence of both bracketed terms in the denominator ensures that this must also be true for the symmetric case of a line from $\mathbf{x} + \mathbf{y}$ at $\phi(\mathbf{x} + \mathbf{y})$ towards \mathbf{x} . It is also easy to introduce a threshold on $|I(\mathbf{x}) - I(\mathbf{x} + \mathbf{y})|$ above which $\Omega_{\mathbf{y}}(\mathbf{x}) = \Omega_{\mathbf{x}}(\mathbf{y}) = 0$.

The overall effect of anisotropy is set by α , which controls the minimum width perpendicular to the main filtering direction when $\gamma(\mathbf{x}) = 1$. Higher values of $\alpha \gg 1$ (or locally lower values of $\gamma(\mathbf{x}) \ll 1$) will lead to $G_{\sigma,\alpha} \Rightarrow G_{\sigma}$. An example application of $G_{\sigma,\alpha}$ is in Fig. 1(c), in which case $\alpha = 0.6$ and σ is set to the same value as in the isotropic Gaussian in Fig. 1(b). The result is similar to iterative anisotropic diffusion [3], but the non-iterative implementation allows the range of the filter to be specifically controlled and removes any possibility of instability. It also has some similarity with another tensor-based method [18], though this was used for image interpolation rather than filtering.

If the anisotropy and direction of the Gaussian filter were constant or changed slowly over I , very efficient implementation techniques involving appropriate shearing of the image [19] could be used. However, in this case $G_{\sigma,\alpha}$ is $\mathcal{O}(l^2)$, hence less efficient than the separable Gaussian G_{σ} which is $\mathcal{O}(l)$, but it can still be implemented in reasonable time. The calculation of γ and ϕ is relatively fast since these are filtered with G_{σ} . Many of the terms in eq. (23) can be pre-calculated across an array covering either the region w or the image I , by judicious use of trigonometric identities, leaving only a small number of simple operations to be performed for each $\{\mathbf{x}, \mathbf{y}\}$ combination. In addition, since $\Omega_{\mathbf{y}}(\mathbf{x}) \equiv \Omega_{\mathbf{y}}(\mathbf{x} + \mathbf{y})$, each weighting calculation can be used twice. It may also

be possible to speed this up further, following the techniques presented in [20].

2) *Improvement of response at corners:* Careful consideration of the example in Fig. 1(c) shows that $G_{\sigma,\alpha}$ introduces a slight diagonal blur where there are strong corners in the image. This is due to the well known problem that the anisotropy γ will be small so long as the eigenvalues of the structure tensor matrix, eq. (18) are the same. This is the desired behaviour in flat regions where they are both small, but is not necessarily beneficial in corners where they can both be large. Hence $\gamma \ll 1$ at the middle of the corner, and also in the surrounding flat region, and the consequent high $\Omega_y(\mathbf{x})$ will cause inappropriate mutual blurring. The prominent diagonal lines of low anisotropy (black) can also be seen across the corners of the tennis racket strings in Fig. 6(c).

This effect can be improved by a slight modification to eq. (20):

$$\gamma(\mathbf{x}) = 1 - \frac{G_{\sigma}(\lambda_2)}{\lambda_1} \quad (25)$$

in which the smaller eigenvalue λ_2 is smoothed to give a more stable reference against which to compare the larger eigenvalue λ_1 . This removes the inappropriate blurring at corners, as seen in Fig. 1(d), by providing a substantially less noisy anisotropy, but with similar spatial resolution, as in Fig. 6(d).

D. Mask shape and orientation

Having defined a robust SV closing and opening, and an appropriately matched linear filter, it remains to choose the particular mask orientation and shape for each pixel in $I(\mathbf{x})$.

1) *Masks from the structure tensor:* The structure tensor has been used in SV opening and closing before [7], and is already the basis of the linear filtering in Section II-C, hence it makes sense to consider this technique first. All that is required is to map γ and ϕ to particular shapes and orientations in the set of masks, e.g. in Fig. 2. The mask shape is defined as:

$$s(\mathbf{x}) = \min \left\{ \left[\left(1 - \frac{\gamma(\mathbf{x})}{0.8} \right) s_{\max} \right], 0 \right\} \quad (26)$$

where $s(\mathbf{x})$ is the shape number, with the thinnest at $s = 0$, as in eq. (8). The mask orientation $o(\mathbf{x})$ is simply that from the possible set, for the shape $s(\mathbf{x})$, which most closely matches $\phi(\mathbf{x})$. Together, $s(\mathbf{x})$ and $o(\mathbf{x})$ define the mask at each location $w(\mathbf{x})$.

2) *Masks from morphological operations:* Since the masks in Section II-D1 are based on orientations from the smoothed structure tensor, they are resistant to noise but do not adjust well to small details in the image. An alternative approach is to trial all possible masks during the initial ranking of both the closing and opening operations, and select the mask whose output centile is closest to the median value in the ranked superset of the union w_u of all the masks, i.e.:

$$w(\mathbf{x}) = w \text{ at } \min_{w \in \text{masks}} \{ |R_{w,c}(I(\mathbf{x})) - R_{w_u,50}(I(\mathbf{x}))| \} \quad (27)$$

This effectively selects whichever mask fits best to the data: and the adjusted centiles in Section II-A2 ensure that fatter masks are preferred if there is no apparent structure in the image, even when corrupted by noise.

The implementation in Section II-A3 means that a large number of masks can be tested whilst incurring only a small processing overhead.

3) *Colour images and optimal combination of masks:* The morphology-based masks have more detail than those based on information from the structure tensor. However, they are poorly defined at signal edges, since no mask, centred on such points, will fit the data well. Fortunately, these are the locations at which the structure tensor defines the orientation very well. A combination of the two methods is hence achieved by starting with the morphological mask definition, then replacing masks with the structure-tensor definition if this gives a thinner shape (smaller s), so long as the anisotropy γ is greater than the noise floor γ_{\min} (see Appendix B). The result of applying this in an SV bitonic filter is in Fig. 1(f).

Morphology-based masks can also improve γ and ϕ used in the linear filter of Section II-C. In this case, eq. (26) is applied in reverse to calculate an effective anisotropy γ_s derived from the morphology-based shape s . If the actual and shape-derived anisotropy are below the expected noise floor, i.e. $\gamma < \gamma_{\min}$ and $\gamma_s < \gamma_{\min}$, then both the anisotropy and orientation are replaced with the shape-derived versions. The linear filter is applied after the SV opening and closing, and hence the masks from the reverse pass in Fig. 3 are used for this purpose.

The simplest way to process colour images is to apply the morphological or linear operations to each colour channel separately. However, using different masks and orientations means that each channel can be smoothed in different directions, resulting in unsightly colour separation. It is hence normally better to define the masks and orientations once from either a grey (for RGB) or lightness (for CIELAB or similar) image, then process all channels individually, but using the same set of masks for all channels. For RGB images, this has the added benefit of reducing the effect of noise on the masks, since the grey image is a reduced-noise average of the RGB channels. In most colour images, structure is largely preserved in the grey version, and the reduction in noise more than compensates for any otherwise small loss of structure.

E. Structurally varying bitonic filter

The overall procedure for implementing the SV bitonic filter is as follows:

- 1) Set the filter radius r and initial centile c_0 (usually 4%). The mask set and centiles follow from eq. (8) and following, and eq. (34).
- 2) Calculate $\gamma(\mathbf{x})$ and $\phi(\mathbf{x})$ for the image $I(\mathbf{x})$ (or the grey version of $I(\mathbf{x})$ if processing a colour image). Also calculate an optimal set of mask shapes $s(\mathbf{x})$ and orientations $o(\mathbf{x})$ by trialling all possible masks in the initial ranking of the opening and closing operations.
- 3) Use $\gamma(\mathbf{x})$ and $\phi(\mathbf{x})$ to improve the masks in $s(\mathbf{x})$ and $o(\mathbf{x})$, following Section II-D3.
- 4) Perform the robust SV opening and closing operations on each image channel separately, eqns. (12) to (15).
- 5) Use $s(\mathbf{x})$ and $o(\mathbf{x})$ to improve the anisotropy and orientations in $\gamma(\mathbf{x})$ and $\phi(\mathbf{x})$, following Section II-D3.

- 6) Calculate the smoothed errors for each channel, ϵ_O and ϵ_C , with eqns. (4) and (5), but with $G_{\sigma,\alpha}$ instead of G_σ , and the anisotropy and orientations $\gamma(\mathbf{x})$ and $\phi(\mathbf{x})$.
- 7) The smoothed errors are combined with the SV opening and closing operations using eq. (7) with $m = 3$.

This is referred to as BitSV, with an example in Fig. 1(f).

A data threshold can also be introduced, as in Section II-B, if the noise level is known or can easily be inferred: this is referred to as BitSVT. If thresholds are used it is also possible to embed BitSVT in a multi-resolution framework, as discussed in Section II-B2.

III. RESULTS

A. Image noise reduction

The fixed bitonic filter was compared to various linear and morphological filters in [1]. Improved performance was clearly demonstrated over the morphological filters, including the Median filter, OCCO filter [21], self-dual area-based grain filters [22], [23], and self-dual levelling based on reconstruction using a Gaussian mask [24], [25]. Since the SV bitonic is a clear improvement on the fixed bitonic, it is instead compared to a range of high-performance linear filters, selected for the breadth of their approaches to noise reduction. In each case l (maximum diameter of the mask for the bitonic) is used to set the parameter which most controls the extent of the filter:

- BM3D** Block-matching¹ [26], with the parameter σ set to a variety of trial values, controlled by l , and centred around the actual standard deviation of the added noise. The profile was left at the default setting, i.e. ‘normal’ ($\sigma < 0.16$), or ‘vn’ ($\sigma \geq 0.16$).
- NLM** Non-local means filter, implemented using a fast algorithm for MATLAB² [4], with the window and search length both set to l , and the filter parameter h set to the standard deviation of the added noise.
- Diffusion** Anisotropic diffusion [27], implemented for MATLAB³, with number of iterations set to l , the integration constant set to the standard deviation of the added noise, the gradient threshold set to twice the standard deviation of the added noise, and the wide-region conduction coefficient.
- Guided** Image-guided filter, implemented using the MATLAB⁴ function `imguidedfilter`⁵ [28], with the local neighbourhood size set to l , and the degree of smoothing set to four times the added noise variance in the image. This is very similar to the well-known bilateral filter [29].
- Anis** The anisotropic Gaussian filter described in section II-C, with $\sigma = 0.33l$ and $\alpha = 0.6$.

- Bitonic** Fixed bitonic filter as in eq. (7), with mask diameter l , and $c = 10\%$.
- BitT** As above, but with added threshold t set to $3.2 \times$ the standard deviation of the noise.
- BitSV** Structurally varying bitonic filter, as described in section II-E, with $c_1 = 4\%$, $\alpha = 0.6$ and mask region of width l .
- BitSVT** As above, but with added threshold t set to $3.2 \times$ the standard deviation of the noise.
- BitMV1** Multi-resolution version of the above, as described in Section II-B2.
- BitMV2** As above, but including the additional optional filter in Fig. 5.

The additional parameters above were set presuming knowledge of the added noise, and chosen for optimal signal to noise ratio (SNR) (and, in the case of Diffusion, reasonable stability) at each noise level. However, only l was optimised over individual images and noise levels for the best SNR and SSIM performance in each case, the other parameters being fixed over all images. The data was extended at the image edges, either symmetrically or by repeating the edge value, with similar results in both cases.

A range of additive Gaussian noise values were tested, giving signal-to-noise ratios (SNR) from ≈ 36 dB (very low noise) to 0 dB (very high noise) in steps of 6 dB, since the relative performance of each filter is expected to vary with noise level. Only Gaussian noise is tested here, in order to sensibly limit the dimensionality of the results. The bitonic filter with fixed mask shapes has previously been tested against various types of noise [1], with similar performance in each case. Structural similarity (SSIM) [30] was included to evaluate image quality after noise reduction as well as increases in SNR.

A reasonable set (given the number of filters, noise levels, and investigated parameters) of 23 images was tested, including standard test images from public-domain sites⁶, various high dynamic range (HDR) images all with the CC0 Creative Commons licence, and two simple computer-generated images. Whilst the intention was for these images to cover a fairly broad range of subjects, they can only be properly considered a convenience sample.

Figure 7 summarises the results over all these images, and all of the noise levels. It reveals the general behaviour of each technique, but does not show the variation across different types of image, nor the characteristic residual patterns in the reduced-noise images. Hence Table I contains detailed results for a selection of these images, with the actual images in Figs. 8 to 11. These figures deliberately cover a wide range of added noise levels, except for very low noise, for which the visual differences are harder to discern. Table I contains some additional results for very high noise levels, with images in Fig. 12.

¹MATLAB BM3D v2.0 software from <http://www.cs.tut.fi/~foi/GCF-BM3D/>

²MATLAB file exchange: Fast Non-Local Means 1D, 2D Color and 3D by Dirk-Jan Kroon, 28 Apr 2010

³MATLAB file exchange: Anisotropic Diffusion (Perona & Malik) by Daniel Lopes, 14 May 2007

⁴MATLAB R2018b, The MathWorks Inc., Natick, MA, US

⁵<http://uk.mathworks.com/help/images/ref/imguidedfilter.html>

TABLE I

SIGNAL TO NOISE RATIO (SNR), STRUCTURAL SIMILARITY (SSIM) AND PROCESSING TIMES FOR IMAGES WITH ADDED GAUSSIAN NOISE σ . THE BEST THREE ALGORITHMS IN EACH CASE ARE HIGHLIGHTED IN BOLD, WITH THE RANK GIVEN BY THE SUPERScript.

Noise σ	0.01			0.04			0.16			0.64		
	SNR dB	SSIM	Time secs	SNR dB	SSIM	Time secs	SNR dB	SSIM	Time secs	SNR dB	SSIM	Time secs
house (256 x 256, grey)												
Original	34.95	0.946	-	23.06	0.590	-	11.04	0.157	-	0.02	0.021	-
BM3D	37.95¹	0.979¹	0.46	31.57¹	0.923¹	0.49	25.60¹	0.831¹	0.61	17.56¹	0.625²	0.56
NLM	36.11	0.969	0.10	30.32²	0.893²	0.26	22.69	0.759	2.45	15.67	0.560	15.59
Diffusion	37.00²	0.970²	0.11	29.01	0.873	0.07	21.49	0.723	0.04	16.68	0.601	0.10
Guided	36.70³	0.968	0.01	28.31	0.866	0.01	20.43	0.507	0.01	13.21	0.166	0.01
Anisotropic	34.95	0.946	0.03	28.60	0.868	0.06	22.60	0.765	0.13	16.26	0.624³	0.26
Bitonic	34.95	0.946	0.06	28.40	0.855	0.05	21.65	0.716	0.06	16.24	0.611	0.07
Bitonic T	36.29	0.969³	0.08	28.24	0.874	0.08	21.62	0.732	0.11	16.68	0.595	0.13
Bitonic SV	35.88	0.962	0.11	29.55	0.878	0.35	23.82	0.797³	1.72	16.89	0.589	1.71
Bitonic SVT	36.28	0.968	0.71	29.59	0.885³	0.59	23.89	0.794	1.69	16.89	0.589	1.81
Bitonic MV1	36.28	0.968	0.69	29.61³	0.885	0.84	24.37³	0.796	0.88	17.36³	0.620	1.43
Bitonic MV2	36.28	0.968	0.70	29.57	0.884	1.46	24.45²	0.809²	1.84	17.41²	0.653¹	4.20
shapes (128 x 128, grey)												
Original	36.61	0.943	-	24.64	0.600	-	12.64	0.190	-	1.73	0.023	-
BM3D	46.88²	1.000¹	0.08	39.00²	0.995¹	0.09	27.35¹	0.912¹	0.13	17.78¹	0.615¹	0.13
NLM	44.91	0.996	0.07	34.70	0.976	0.33	22.49	0.751	0.33	16.77	0.543	3.92
Diffusion	41.93	0.984	0.05	34.41	0.974	0.04	22.10	0.732	0.03	17.53³	0.592³	0.05
Guided	43.22	0.993	0.01	30.88	0.898	0.01	21.42	0.526	0.01	14.49	0.179	0.03
Anisotropic	36.61	0.943	0.03	29.18	0.928	0.03	23.46	0.800	0.04	17.06	0.608²	0.07
Bitonic	40.78	0.988	0.03	30.61	0.916	0.02	22.01	0.715	0.03	17.11	0.586	0.03
Bitonic T	50.39¹	0.999²	0.06	38.80	0.988³	0.08	22.11	0.721	0.07	17.42	0.572	0.05
Bitonic SV	44.67	0.996	0.11	35.25	0.977	0.16	25.41	0.872	0.28	17.16	0.512	0.42
Bitonic SVT	45.21	0.997	0.10	38.21	0.985	0.36	25.45	0.872	0.34	17.16	0.512	0.51
Bitonic MV1	44.97	0.996	0.12	38.80³	0.986	0.46	25.95³	0.888³	0.44	17.29	0.532	0.56
Bitonic MV2	46.59³	0.998³	0.13	40.09¹	0.993²	0.80	26.60²	0.910²	0.56	17.74²	0.577	0.46
blue rocks (730 x 1024, colour)												
Original	34.31	0.984	-	22.42	0.817	-	10.50	0.343	-	-0.58	0.059	-
BM3D	37.24¹	0.993¹	8.84	29.22¹	0.957¹	9.04	22.76¹	0.839¹	18.44	15.81³	0.622¹	18.88
NLM	35.22	0.987	7.69	26.93	0.938²	7.49	21.36³	0.782	41.16	14.76	0.560	369.29
Diffusion	35.70²	0.990²	7.18	26.69	0.927	2.82	19.84	0.738	1.45	15.05	0.588	7.25
Guided	35.62³	0.988	0.34	26.56	0.925	0.23	19.05	0.691	0.24	12.22	0.310	0.50
Anisotropic	34.31	0.984	0.81	25.70	0.912	0.94	20.88	0.783	2.63	15.31	0.600	4.48
Bitonic	34.31	0.984	1.74	24.86	0.893	2.12	19.59	0.731	2.62	15.14	0.587	2.36
Bitonic T	35.09	0.989	2.78	26.10	0.922	2.24	19.83	0.740	2.47	15.13	0.586	2.38
Bitonic SV	34.31	0.984	3.61	26.12	0.918	5.82	21.14	0.796	10.99	15.41	0.601	24.56
Bitonic SVT	35.28	0.989³	6.74	27.00³	0.936	8.62	21.20	0.799	11.96	15.41	0.601	24.51
Bitonic MV1	35.24	0.989	8.13	27.03²	0.936	11.01	21.32	0.802³	12.26	15.85²	0.610³	20.58
Bitonic MV2	35.19	0.989	14.75	26.89	0.937³	15.91	21.38²	0.802²	19.98	15.96¹	0.619²	34.76
south sound (680 x 1024, colour)												
Original	34.59	0.974	-	22.76	0.720	-	10.74	0.180	-	-0.37	0.023	-
BM3D	40.63¹	0.990¹	9.81	34.87¹	0.968¹	10.93	29.07¹	0.937¹	18.51	19.32³	0.832	12.17
NLM	38.34³	0.985	7.09	32.84²	0.962²	74.91	26.81	0.912	132.89	18.87	0.816	538.52
Diffusion	37.81	0.988²	6.87	31.69	0.958	5.16	24.68	0.895	3.55	18.67	0.829	7.36
Guided	38.51²	0.986³	0.25	30.63	0.948	0.23	22.07	0.728	0.27	13.88	0.281	0.29
Anisotropic	36.76	0.985	0.39	31.13	0.954	0.97	25.90	0.910	2.48	19.12	0.849²	6.46
Bitonic	36.61	0.983	1.56	30.55	0.951	1.98	24.81	0.895	2.11	19.04	0.845	3.17
Bitonic T	37.77	0.986	2.13	31.09	0.955	2.31	25.05	0.896	2.42	18.81	0.831	2.66
Bitonic SV	36.84	0.984	3.35	31.88	0.957	6.86	26.62	0.918	18.20	19.27	0.844	28.90
Bitonic SVT	37.68	0.985	6.06	31.91	0.958	9.22	26.59	0.917	18.82	19.27	0.844	31.96
Bitonic MV1	37.59	0.984	7.22	31.97	0.958	9.79	27.05³	0.918³	17.79	19.84²	0.848³	31.19
Bitonic MV2	37.57	0.984	13.05	32.09³	0.958³	20.39	27.09²	0.921²	35.33	19.90¹	0.857¹	67.66

TABLE II
SIGNAL TO NOISE RATIO (SNR), STRUCTURAL SIMILARITY (SSIM) AND PROCESSING TIMES FOR IMAGES WITH VERY HIGH NOISE. THE BEST THREE ALGORITHMS IN EACH CASE ARE HIGHLIGHTED IN BOLD, WITH THE RANK GIVEN BY THE SUPERScript.

Image	fruits (512 x 512)			peppers (512 x 512)			tulips (512 x 768)			marina bay (600 x 900)		
	SNR dB	SSIM	Time secs	SNR dB	SSIM	Time secs	SNR dB	SSIM	Time secs	SNR dB	SSIM	Time secs
Original	1.34	0.043	-	-0.93	0.042	-	-1.27	0.062	-	-1.66	0.066	-
BM3D	18.55	0.685	4.37	16.03 ³	0.695 ³	4.34	14.93 ²	0.621 ²	7.65	13.78 ¹	0.636 ¹	8.83
NLM	17.55	0.651	192.48	15.21	0.640	79.76	13.98	0.541	41.44	12.86	0.583	263.40
Diffusion	17.64	0.671	2.48	15.37	0.677	2.72	14.03	0.594	1.99	13.18	0.610	5.47
Guided	14.62	0.297	0.14	12.15	0.326	0.17	11.44	0.355	0.29	10.83	0.312	0.22
Anisotropic	18.17	0.684	1.70	15.61	0.683	1.57	14.26	0.599	1.92	13.38	0.620	2.57
Bitonic	17.69	0.671	0.70	15.63	0.678	0.62	14.21	0.592	0.92	13.20	0.609	1.39
Bitonic T	17.80	0.671	0.79	15.59	0.677	0.70	14.16	0.590	1.03	13.17	0.609	1.41
Bitonic SV	18.61 ³	0.687 ³	8.35	15.92	0.689	7.30	14.49	0.606	7.58	13.34	0.620	16.28
Bitonic SVT	18.61 ³	0.687 ³	8.40	15.92	0.689	7.28	14.49	0.606	7.57	13.34	0.620	16.20
Bitonic MV1	18.99 ²	0.691 ²	8.10	16.44 ²	0.699 ²	7.06	14.89 ³	0.613 ³	9.70	13.67 ³	0.626 ³	13.73
Bitonic MV2	19.14 ¹	0.699 ¹	13.89	16.61 ¹	0.711 ¹	11.78	15.02 ¹	0.626 ¹	14.11	13.73 ²	0.634 ²	22.90

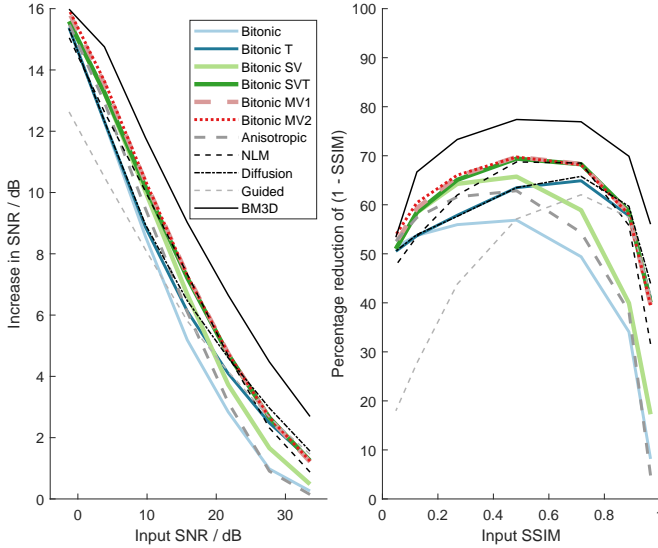


Fig. 7. Summarised results over the complete set of 23 images. Optimal filter performance (after optimisation of filter extent l in each case) is averaged over each noise level in each image. Increase in SNR is shown on the left, and percentage reduction of residual SSIM (i.e. the difference from the ideal SSIM value of 1) on the right.

B. Processing times

The processing times for the various algorithms are included in Table I, for whichever value of l generated the best SNR and SSIM results, measured using an i7-7500 CPU at 2.7 GHz. The BitSV algorithm is of particular interest, since both the anisotropic filter $G_{\sigma,\alpha}$ and the implementation of robust SV opening and closing are novel. Figure 13 contains more details on the processing times for these algorithms, all applied to a 512×512 colour image. Fixed and robust SV openings are first considered, both for histogram-based and sorting-based

implementations, since whilst the former is more efficient, the latter is necessary if the data is not of integer type. The fixed opening is also implemented in the SV framework, revealing the roughly 3 to 4 \times overhead purely due to the framework, rather than the increased number of trial masks. Trials of up to 32 masks (for $l = 21$) for the SV version introduce a fairly small overhead.

Both implementations of the key parts of the SV bitonic are considered: the overall algorithm is $\mathcal{O}(l^{1.5})$ for the histogram-based version and $\mathcal{O}(l^2)$ for the sorting-based version. Note that both of these performances are acceptable, since l is the filter width, not the number of elements in the filter mask, which is proportional to l^2 . All filters in Fig. 13 are based on local windows, and hence also scale with the number of pixels in the signal $I(\mathbf{x})$.

IV. DISCUSSION

Development of the SV bitonic filter was motivated by the novelty of differentiating signal and noise via bitonicity, and the apparent gain from allowing the mask shape to vary with the structure of the signal in an image. Figure 7 shows that this is vindicated by the performance of BitSV, which is better than the original Bitonic for both SNR and SSIM, at all noise levels, usually by a considerable margin. This is particularly clear in Fig. 11 and for the blue rocks image in Fig. 10 with very high noise. The additional performance does come at an increased, though still competitive, processing time. In the previous work [1], Bitonic was only better than NLM and Diffusion in a few specific cases, most notably for varying noise, whereas BitSV now outperforms both of these linear alternatives for medium to high noise levels, producing distinctly sharper images, as in Fig. 11.

In contrast, the addition of a data threshold to both Bitonic and BitSV has a strong effect on the performance in lower noise scenarios. This is apparent in the numerical results in Table I, but particularly clear from the SSIM results in Fig. 7. Both BitT and BitSVT offer much better SSIM improvement

⁶<https://homepages.cae.wisc.edu/~ece533/images/> and <http://decsai.ugr.es/cvg/CG/base.htm>

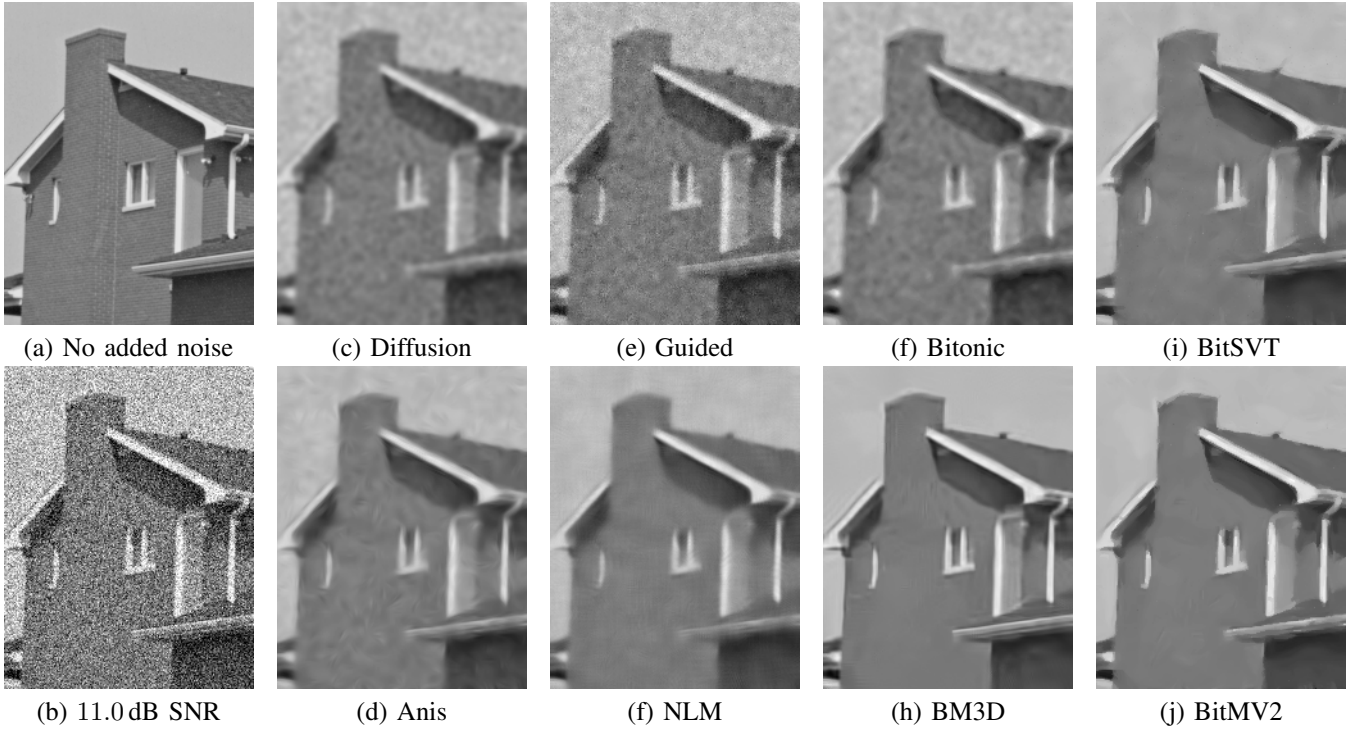


Fig. 8. Results for noise in (b) added to section of 'house' image in (a): see Table I for further details.

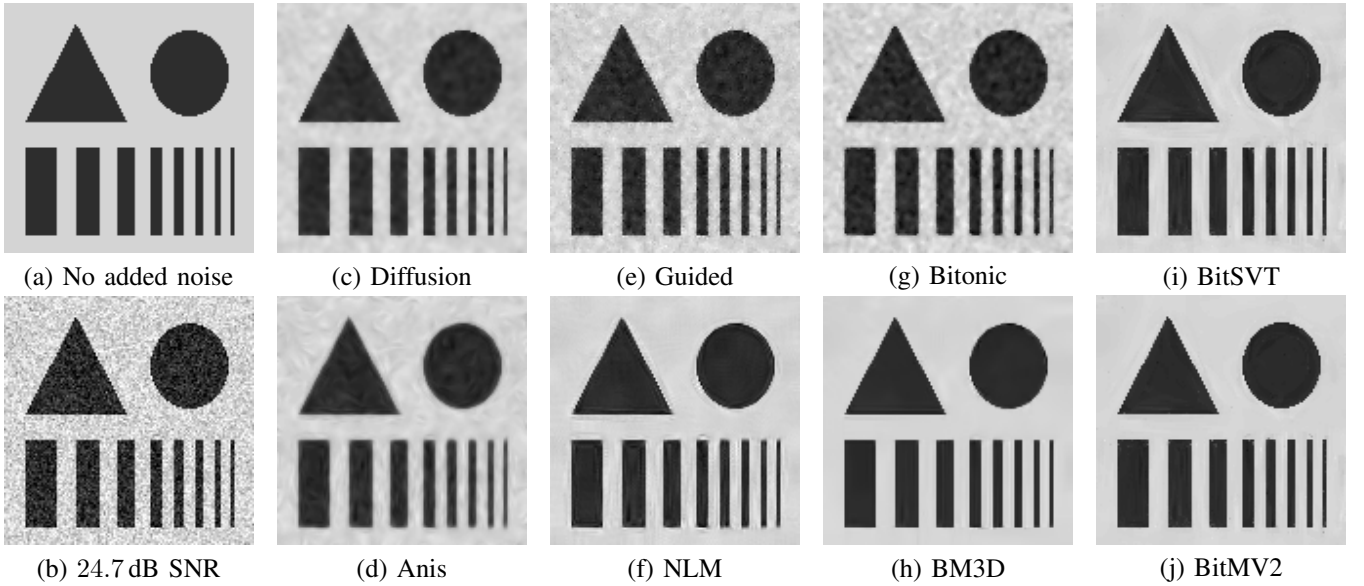


Fig. 9. Results for noise in (b) added to 'shapes' image in (a): see Table I for further details.

at low noise than their non-thresholded counterparts, but tend to the same performance at higher noise. BitSVT outperforms NLM at all noise levels, and generates less distracting characteristic patterns. This is clear in the synthetic shapes image in Fig. 9 but is also apparent in natural images. In these examples Diffusion also generates a visually pleasing result, though both this and Guided fail to do so at most other noise levels. Guided is, however, extremely fast, and does have good performance at the lowest noise levels.

The improvements in BitSVT are not, however, sufficient to achieve better SNR and SSIM results than BM3D, which is the strongest performer, except at the highest noise levels.

Here, the addition of the BitMV2 multi-resolution framework to BitSVT makes it comparable to BM3D (Tables I and II), though at the cost of doubling the processing time. In contrast, BitMV1 offers a slight improvement over BitSVT at most noise levels, for minimal increase in processing time: in fact BitMV1 often requires a smaller mask size than BitSVT, in which case the processing time can even be reduced.

The numerical results are not an adequate summary of the performance, since the characteristic residual noise from each algorithm is very different, even for similar SNR or SSIM values. These characteristics can only be seen in images: they are very visible on the house in Fig. 8. At these noise levels,

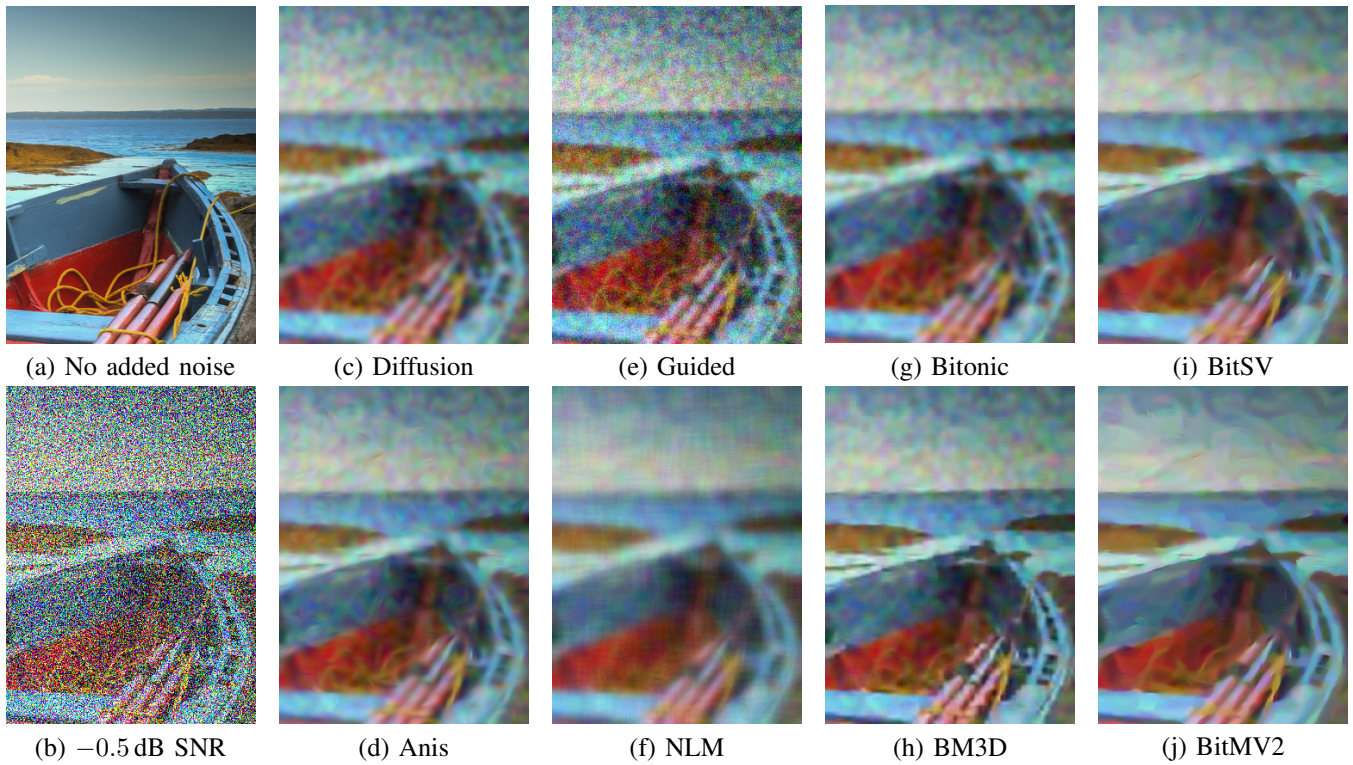


Fig. 10. Results for noise in (b) added to section of ‘blue rocks’ image in (a): see Table I for further details.

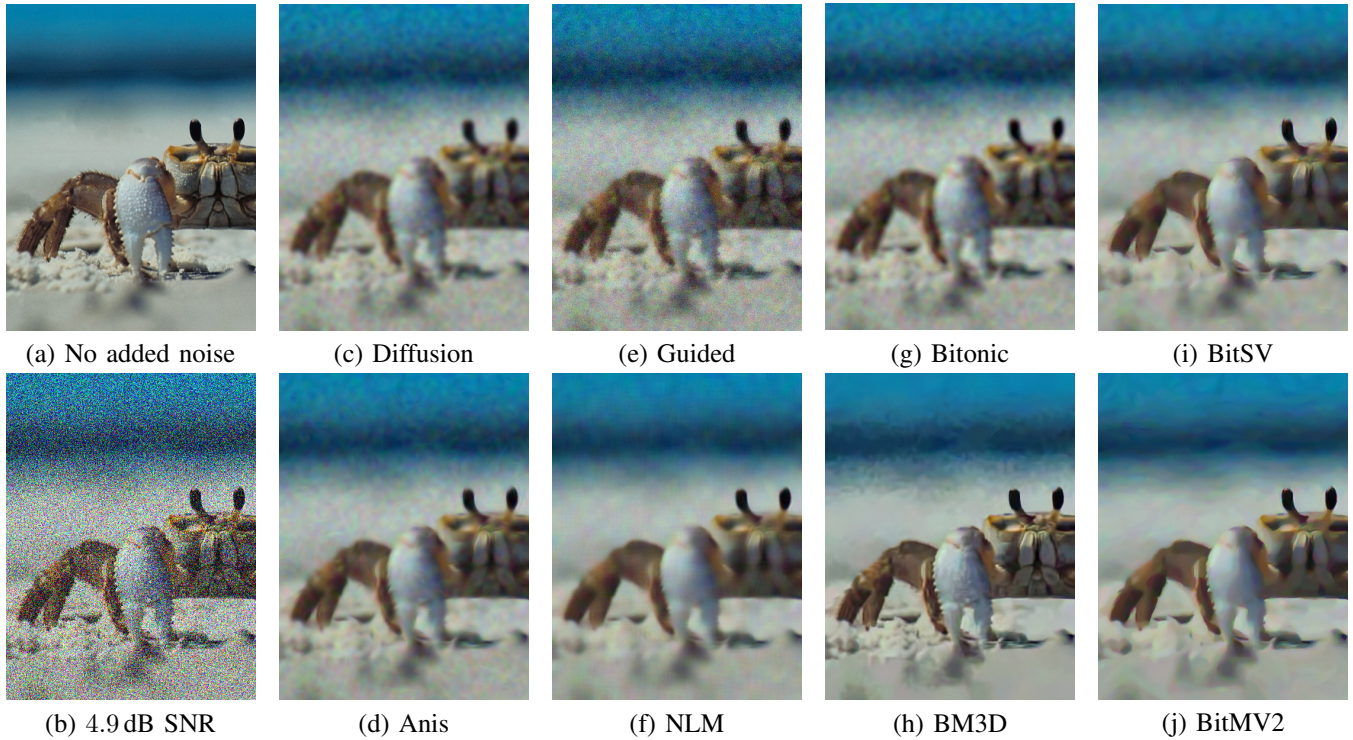


Fig. 11. Results for noise in (b) added to section of ‘south sound’ image in (a): see Table I for further details.

Diffusion blurs the signal as well as reducing noise, and the data threshold in Guided either leads to much noise surviving (as shown) or too much blurring. Bitonic has sharper edges than Diffusion but BitSVT is a very clear improvement. NLM starts to reveal a cross-hatching which results from finding false patterns in the noise, and BM3D also reveals some false

patterning across the front face of the house. BitMV2 has less characteristic noise, though the mask shape (particularly thin ellipses) is evident in some places; however there is also slightly less detail than with BM3D. At a slightly higher noise level, the south sound image in Fig. 11 also demonstrates the contrast to BM3D. Details of the crab claw and underside are

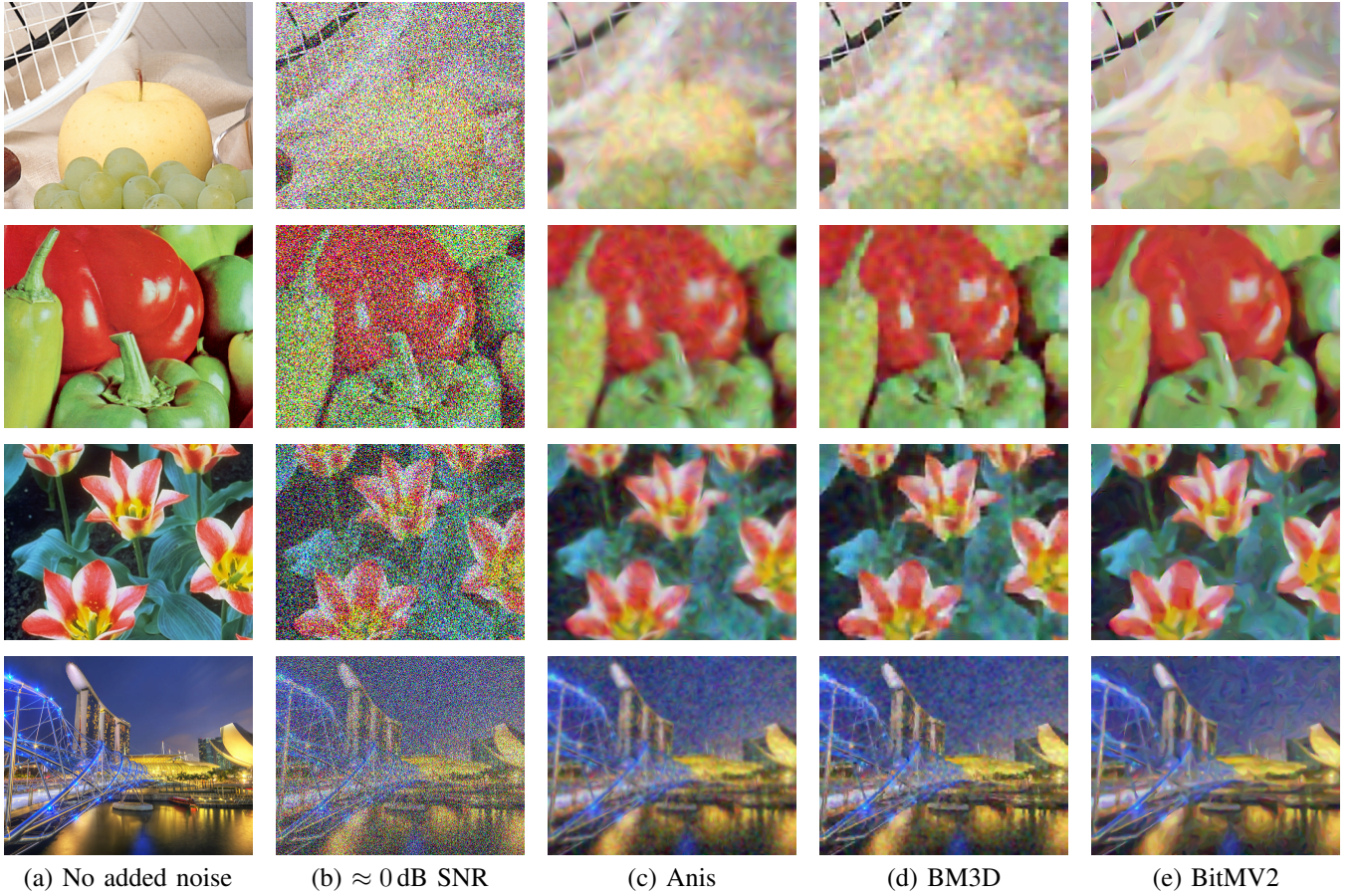


Fig. 12. Additional results for very high noise in images, from top to bottom: ‘fruits’, ‘peppers’, ‘tulips’ and ‘marina bay’. See Table II for further details.

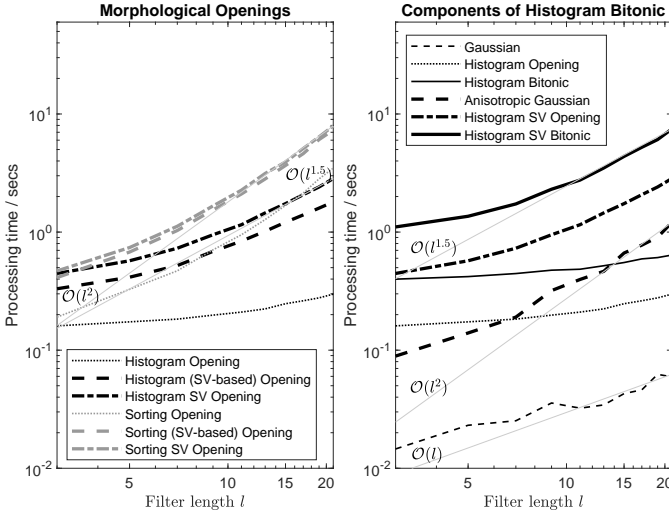


Fig. 13. Processing times for the SV bitonic filter. These are for a single 512×512 colour image, based on the average of ten results. On the left is for just the morphological opening. The dashed lines show a fixed (single mask) filter, but with the SV framework: there is a considerable overhead in storing region indices and checking for mask locations. However the subsequent trialling of up to 32 masks (dash-dot lines) is much less significant. On the right are the components of the histogram-based bitonic and SV bitonic: whilst the anisotropic Gaussian is much slower than the Gaussian, it is still the opening and closing operations that dominate the time taken for both the bitonic filters. Sorting-based versions of the bitonic are slower, tending to $\mathcal{O}(l^2)$ rather than $\approx \mathcal{O}(l^{1.5})$, in the mask width l .

preserved since the patterns are sufficiently repeated: BitMV2 shows less detail, but the edges are very clear and there is little characteristic noise across the smooth background. This is a direct result of enforcing bitonicity across a larger scale due to the increased noise, but similar features are still visible in cases with medium noise levels.

The very highest noise levels, as in Fig. 12 and the blue rocks image in Fig. 10, reveal the block-based structure on which BM3D is based, apparent at the oars and along the edge of the boat. At this level, NLM has poor performance, and prohibitively long processing times. BitMV2, whilst not preserving very small features, otherwise performs well, recovering edges smoothly with limited characteristic noise across more constant regions. Performance at high noise levels is arguably the strongest characteristic of BitMV2.

The BitSV has required several novel developments with potentially broader applications. These include an efficient implementation of robust SV opening and closing, with a framework that allows many masks to be tested. Together with the careful analysis of relative centiles for different shapes, this has enabled mask choice to be determined from morphological operations, rather than from a smoothed structure tensor. The local anisotropy derived from such a tensor has also been improved at corner locations. Good noise-reduction results have been achieved by using this robust SV with simple elliptical masks, but the framework also opens up the possibility of investigating more complex mask shapes, which may be what

is needed to preserve smaller details in the images.

A new formulation has also been presented for non-iterative directional Gaussian smoothing, which can optionally also be thresholded at no additional cost to processing times. Whilst this technique alone does not perform as well as when embedded in BitSV, it still has quite reasonable noise-reduction capability at high noise levels (Table II and Fig. 12), and is faster than most other options.

V. CONCLUSION

The structurally varying bitonic is a considerable improvement on the fixed bitonic for image noise reduction, whilst increasing the processing time by usually less than an order of magnitude. It is increasingly competitive with linear noise-reduction algorithms, outperforming non-local means at all noise levels and the block-matching 3D filter for very high noise. In such scenarios, the structurally varying bitonic has less characteristic residual noise and very good edge definition, with similar processing times, though with some loss of small scale detail compared to block-matching. The non-iterative directional Gaussian used in the new bitonic filter is much faster and yet also has good performance in high noise. The novel implementation of robust structurally varying opening and closing operations enables future development of the bitonic, and morphological operations more generally, with more complex mask shapes. Inclusion of data thresholding has improved performance in low noise and enables the adoption of a multi-resolution framework around morphological operations for high noise.

Implementations of all the novel filters in this paper are available for Matlab⁷ and also for Windows in wxDicom⁸ software.

APPENDIX

A. Experimental basis for equivalent mask centiles

For SV ranking operations applied to pure noise, ideally each fatter shape (larger s) should return the same ranked intensity (on average) as when using the centile c_0 for the thinnest shape ($s = 0$). The returned ranking from eq. (27) is the closest to the median over all orientations, so fatter shapes with fewer orientations need a higher centile to compensate. Figure 14 (left) shows experimental results for this additional centile, for c_0 from 0% to 50% and for Gaussian noise, though the distribution of the noise is not important since the results are only affected by the ranking of the data, not the data values.

The additional centile is a function of c_0 , and is also highly dependent on the specific shapes, relative pixel overlap, and number of orientations, which can be summarised by the parameters a and b :

$$a = \sqrt{\frac{|w_0|}{|w_s|}}, \quad b = \sqrt{\frac{|w_0|}{|w_{s_{\max}}|}} \quad (28)$$

where the deviation of the centile output is expected to vary approximately with the square-root of the number of elements

⁷MATLAB file exchange: Structurally Varying Bitonic Filter by Graham Treece, 16 Aug 2018

⁸http://mi.eng.cam.ac.uk/Main/GMT_wxDicom

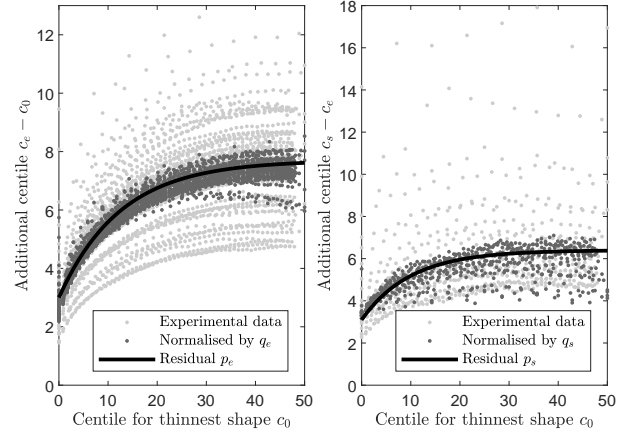


Fig. 14. Experimental data for equivalent centiles. The graph on the left shows the results of ranking noise with eq. (12) and all mask sets with radii 1 to 20. The x-axis shows the centile c_0 for the thinnest shape, and the y-axis the additional centile required for each other shape in the mask set to return the same ranked value. The graph on the right is a similar test, showing the additional centile over and above this, required to ensure that the circular mask will always be used for pure noise 95% of the time. In both cases, each grey dot shows the average of 2000 experiments for one shape s , radius r and centile c_0 , black dots are normalised by q , and the solid line is the residual dependency p .

$|w|$ in each mask. The dependency with a and b , and the residual dependency with c_0 , are found experimentally, such that the adjusted centile c_e is given by:

$$q_e = 2(1-a) \left(\frac{\sqrt{b}}{1-b} \right) \quad (29)$$

$$p_e = 3 + 4.7e^{-8c_0} \quad (30)$$

$$c_e = c_0 + p_e q_e \quad (31)$$

Figure 14 (left) shows the experimental data normalised by q_e , and the residual dependency p_e as a bold line.

Use of the centile c_e in eq. (31) would result in an equal chance of fitting a mask of any shape if there is no structure in the image. However a fatter mask would be preferred for pure noise, since the larger number of elements in the mask improve noise reduction. Hence an additional positive offset is added to c_e for fatter mask shapes, which ensures that thinner masks are only used where this is justified by the data. Figure 14 (right) shows the required additional offset, found by matching the 95% centile of 2000 results for the ranked value of a circular mask ($s = s_{\max}$) to the ranked value for the thinnest mask. This offset is larger for smaller masks, since lower $|w_{s_{\max}}|$ introduces more variability into the ranking results when applied to noise. This, and the residual dependency with c_0 , are again found experimentally, such that the final centile c_s is given by:

$$q_s = \frac{25}{\sqrt{|w_{s_{\max}}|}} \quad (32)$$

$$p_s = 3.1 + 3.3e^{-10c_0} \quad (33)$$

$$c_s = c_e + p_s q_s \frac{s}{s_{\max}} \quad (34)$$

Figure 14 (right) shows the experimental data normalised by q_s , and the residual dependency p_s as a bold line.

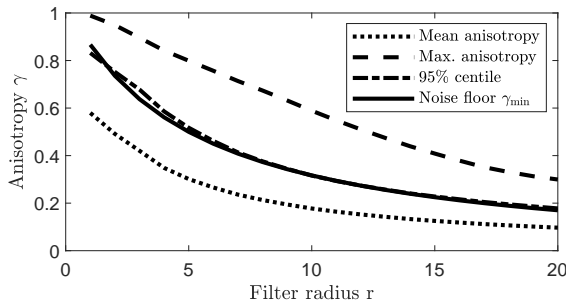


Fig. 15. Experimental data for anisotropy noise floor. The graph shows the mean, maximum and 95% centile for the anisotropy γ when using eq. (20) on a 1000×1000 image of Gaussian noise, with varying filter radii r .

B. Experimental basis for anisotropy noise floor

The image anisotropy γ , calculated using eq. (20), will give a non-zero value even when the image just contains noise. It is hence necessary to establish a lower limit for γ below which neither the anisotropy nor the corresponding orientation ϕ can be trusted. This lower limit γ_{\min} is experimentally derived, as shown in Fig. 15. An approximate function is fitted to the 95% centile of this data, which gives a high probability that anisotropy above this value will represent genuine structure in the image:

$$\gamma_{\min} = \frac{5.5}{r + 5} - 0.05 \quad (35)$$

where γ_{\min} reduces with increasing filter radius r , as expected.

REFERENCES

- [1] G. M. Treece, "The bitonic filter: linear filtering in an edge-preserving morphological framework," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5199–5211, Nov. 2016.
- [2] —, "Refinement of clinical x-ray computed tomography (CT) scans containing metal implants," *Comput. Med. Imaging Graph.*, vol. 56, pp. 11–23, Feb. 2017.
- [3] J. Weickert, *Anisotropic diffusion in image processing*. Teubner Stuttgart, 1998, vol. 1.
- [4] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.
- [5] V. Čurić, A. Landström, M. J. Thurley, and C. L. L. Hendriks, "Adaptive mathematical morphology—a survey of the field," *Pattern Recognition Letters*, vol. 47, pp. 18–28, 2014.
- [6] N. Bouaynaya and D. Schonfeld, "Theoretical foundations of spatially-variant mathematical morphology part ii: Gray-level images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 837–850, 2008.
- [7] A. Landström and M. J. Thurley, "Adaptive morphology using tensor-based elliptical structuring elements," *Pattern Recognition Letters*, vol. 34, no. 12, pp. 1416–1422, 2013.
- [8] P. Soille and H. Talbot, "Directional morphological filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1313–1329, 2001.
- [9] O. Tankyevych, H. Talbot, P. Dokládál, and N. Passat, "Spatially-variant morpho-hessian filter: Efficient implementation and application," in *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer, 2009, pp. 137–148.
- [10] R. Verd-Monedero, J. Angulo, and J. Serra, "Anisotropic morphological filters with spatially-variant structuring elements based on image-dependent gradient fields," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 200–212, 2011.
- [11] M. Kass and J. Solomon, "Smoothed local histogram filters," *ACM Trans. Graph.*, vol. 29, no. 4, p. Article 100, 2010.
- [12] S. Perreault and P. Hébert, "Median filtering in constant time," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2389–2394, 2007.

- [13] S. Paris, S. W. Hasinoff, and J. Kautz, "Local laplacian filters: Edge-aware image processing with a laplacian pyramid," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 68–1, 2011.
- [14] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A multigrid tutorial*, 2nd ed. SIAM, 2000.
- [15] E. Catmull and R. Rom, "A class of local interpolating splines," in *Comput. Aided Geom. Des.*, R. Barnhill and R. Riesenfeld, Eds. San Francisco: Academic Press, 1974, pp. 317–326.
- [16] L. J. Van Vliet and P. W. Verbeek, "Estimators for orientation and anisotropy in digitized images," in *ASCI*, vol. 95, 1995, pp. 16–18.
- [17] W. C. Wong, A. C. Chung, and S. C. Yu, "Trilateral filtering for biomedical images," in *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, 2004, pp. 820–823.
- [18] A. Baghaie and Z. Yu, "Structure tensor based image interpolation method," *AEÜ - Int. J. Electron. Commun.*, vol. 69, no. 2, pp. 515–522, 2015.
- [19] C. H. Lampert and O. Wirjadi, "An optimal nonorthogonal separation of the anisotropic gaussian convolution filter," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3501–3513, 2006.
- [20] J. Baek and D. E. Jacobs, "Accelerating spatially varying gaussian filters," *ACM Trans. Graph.*, vol. 29, no. 6, p. 169, 2010.
- [21] E. Aptoula and S. Lefevre, "A comparative study on multivariate mathematical morphology," *Pattern Recognition*, vol. 40, no. 11, pp. 2914–2929, 2007.
- [22] V. Caselles and P. Monasse, "Grain filters," *J. Math. Imaging Vis.*, vol. 17, no. 3, pp. 249–270, 2002.
- [23] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 860–872, 2000.
- [24] P. Maragos and G. Evangelopoulos, "Leveling cartoons, texture energy markers, and image decomposition," in *Proceedings of the Eighth International Symposium on Mathematical Morphology*, ser. ISMM '07. Kluwer Academic Publishers, 2007, pp. 125–138.
- [25] J. Serra, C. Vachier, and F. Meyer, "Levelings," in *Mathematical Morphology: From Theory to Applications*. Wiley, 2013, pp. 199–228.
- [26] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [27] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, 1990.
- [28] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, 2013.
- [29] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision*, 1998. IEEE, 1998, pp. 839–846.
- [30] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.



Graham Treece obtained an MA degree in engineering, with distinction, from the University of Cambridge, UK in 1994. Following a few years working as a software engineer in noise cancellation and radio communications, he returned to the University of Cambridge and completed his PhD in 3D ultrasound in 2001. He continued to pursue ultrasound research at Cambridge during a five year research fellowship from the Royal Academy of Engineering, focusing on the measurement of attenuation and elastography. From 2008 to 2015 he was the inaugural Evelyn Trust lecturer in Engineering for Clinical Practice in Cambridge, with research spanning most medical imaging modalities. His recent research has mainly concerned X-ray Computed Tomography, developing Cortical Bone Mapping (CBM) for the detailed analysis and visualisation of bone diseases, and also Refined Metal Artefact Reduction (RMAR) for reducing noise due to the presence of metal. He is currently a reader in information engineering at the University of Cambridge. He is an author of a variety of free software packages, including Stradwin (ultrasound acquisition, medical data visualisation and CBM) and wxDicom (image processing and RMAR).